A metaheuristic method to design one-way electric car sharing system

Taki Eddine Harbi[1], Ben Lauwens[1]
[1]Royal Military Academy, Belgium
✉ Taki.Harbi@mil.be

1. Context

(Electric) Car Sharing ECS: A sustainable mobility solution where (electric) vehicles are made accessible to the public on a short-term basis. Users can conveniently rent and return vehicles through a digital platform.

Types of car sharing system:

- *Round Trip Car Sharing*: Users pick up and return the vehicle to the same station making it ideal for planned, round-trip journeys.
- *One-Way Car Sharing*: Users can pick up a vehicle from one station and drop it off at another, providing flexibility for point-to-point travel without the need for a return trip.
- *Free-Floating Car Sharing*: Vehicles are dispersed throughout a defined service area (a city), allowing users to locate, access, and drop off cars at any legal parking spot within the operational zone, providing maximum flexibility.

Strategic decisions to increase the profit:

- *Stations' locations:* generally, choose from several potential locations.
- *Fleet size:* number and type of cars to purchase.
- *cars deployment :* number of cars at each station.

2. One-way ECS: problem definition

Given:

- List of requests K. Each $k \in K$ consists of origin O_k and destination D_k, and it can have multiple trips (see figure)
- Set of potential stations J, each has a capacity c_j.

Decision variables for one-way car sharing:

- Select subset of locations $\bar{J} \subset J$ to establish stations.
- For each selected station, set the number of cars L_j.
- For each $k \in K$, select a feasible trip. ($\bar{K}$: list of assigned requests)

Objective: $\boldsymbol{max} \left(\sum_{k \in K} \boldsymbol{revenue(k)} - \sum_{j \in \bar{J}} \boldsymbol{station\_cost(j)} - \sum_{j \in \bar{J}} \boldsymbol{L_j} * \boldsymbol{car\_cost}\right)$

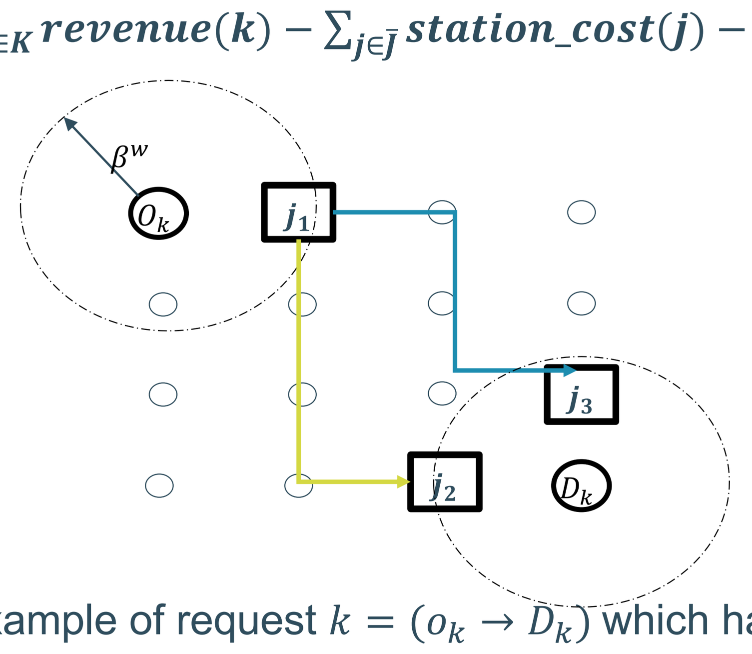


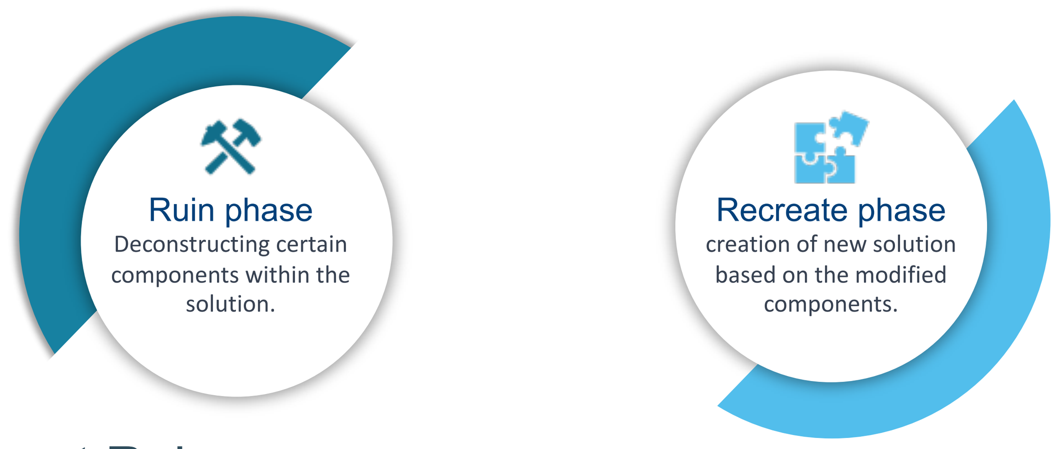
Fig1: Example of request $k = (o_k \to D_k)$ which has 2 feasible trips

3. Solution Approach: Simulation Annealing

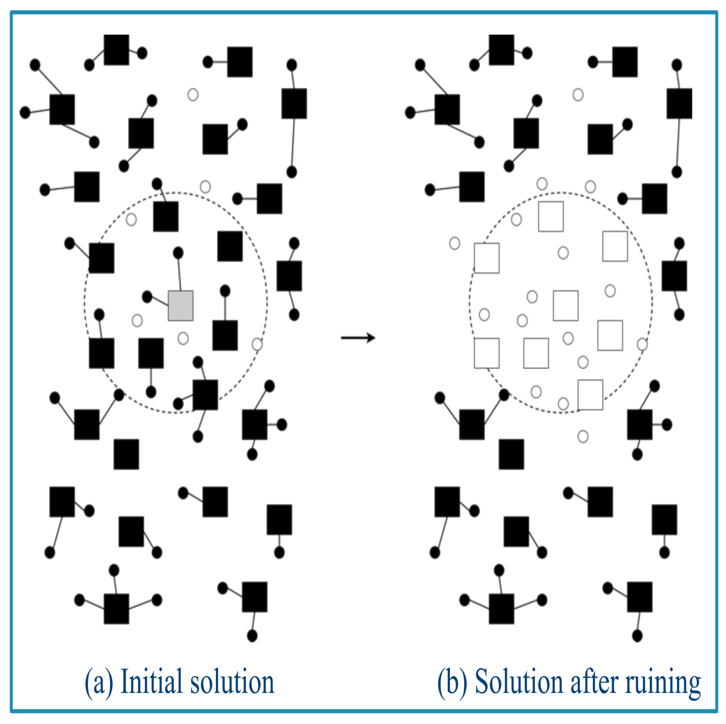
```
procedure SIMULATED ANNEALING(s0, T0, Tf, α, I)
    s, fit ← s0, OBJECTIVE_FUNC(s0)
    best_s, best_fit, T ← s, s_fit, T0
    while T > Tf do
        for I iterations do
            neigh_s ← SAMPLE_NEIGHBORHOOD(s)
            neigh_fit ← OBJECTIVE_FUNC(neigh_s)
            if neigh_fit < s_fit then
                s, s_fit ← neigh_s, neigh_fit
                if neigh_fit < best_fit then
                    best_s, best_fit ← neigh_s, neigh_fit
            else ACCEPT_WITH_PROBABILITY( exp Δfit/T )
        T ← T * α
```

- Inspired from the steel heating process.
- Parameters: initial solution s_0; initial and final temperatures T_0 and T_f; cooling rate α and number to of iterations to spend at each temperature.
- Solution = stations' locations + cars deployments + paths for the requests.
- Accept worse solutions to escape from local optimal solutions.
- Explore the solutions search space using **Ruin and Recreate (R&R)** procedure.

4. Adjacent R&R procedure

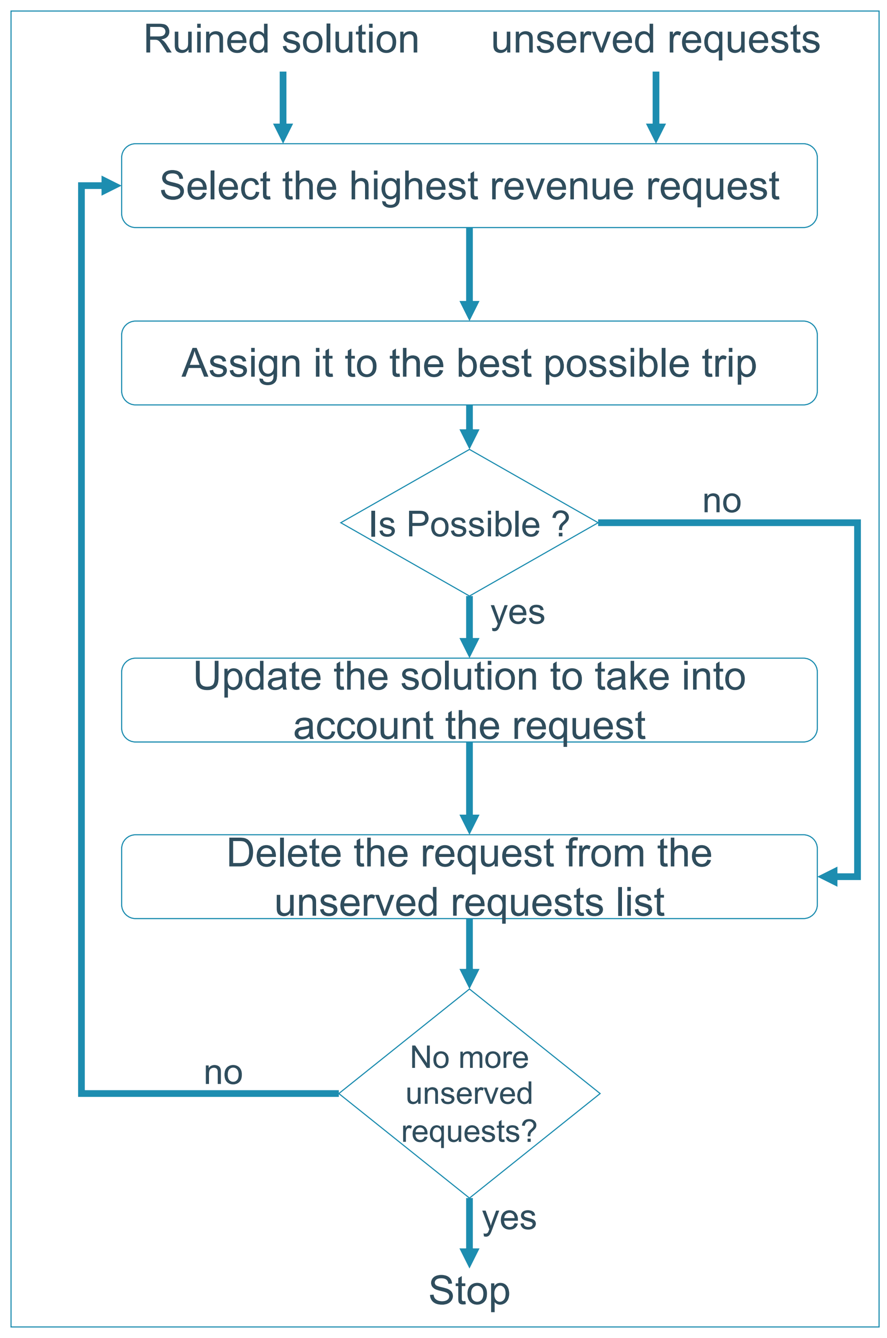


4.1 Adjacent Ruin



- Choose a random station to serve as the starting point (grey square).
- Close stations that are geographically nearby.
- Number of stations to close N_S:
  $N_S^{max} = ruin\_depth * |open\_stations(s)|$
  $N_s = \lfloor U(1, N_s^{max} + 1) \rfloor$

4.1 Greedy Recreate



5. Results

Instances:

- Scenarios taken from [1] based on a real data obtained from Manhattan (New York, USA) taxi trips, with total of 85 potential station.
- **Experiment environment**: Julia 1.9 on Apple M1 CPU, 8 cores and 8GB of RAM.

SA parameter: $T_0 = 1000, T_f = 1, \alpha = 0.95, I = 100, \beta = 0.5$

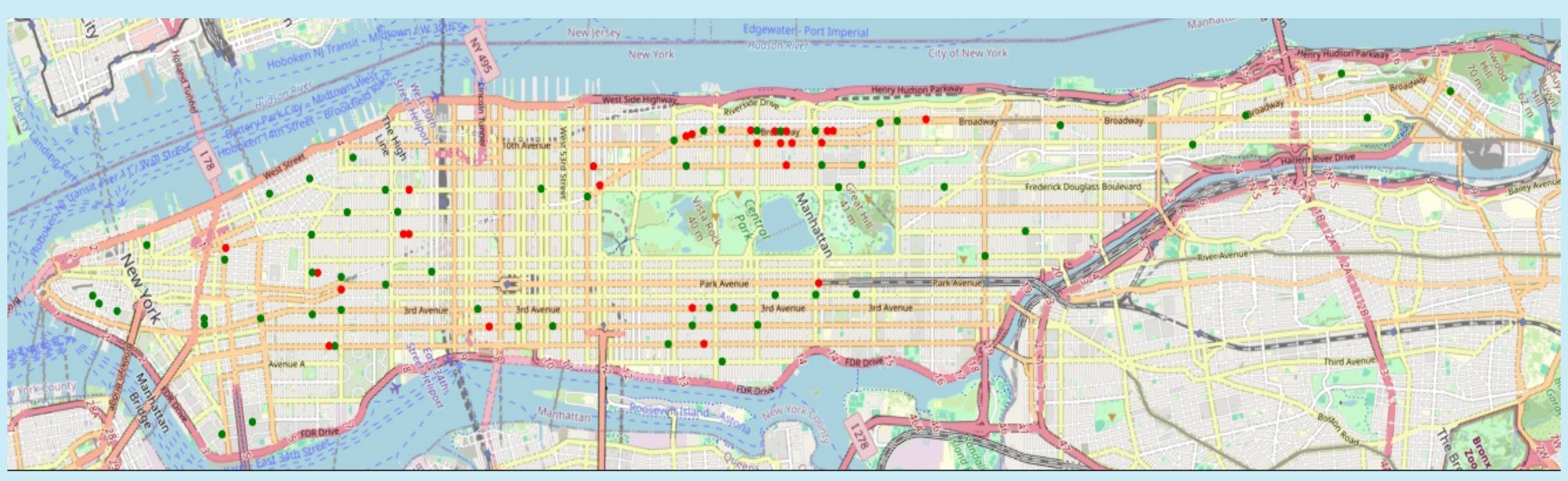
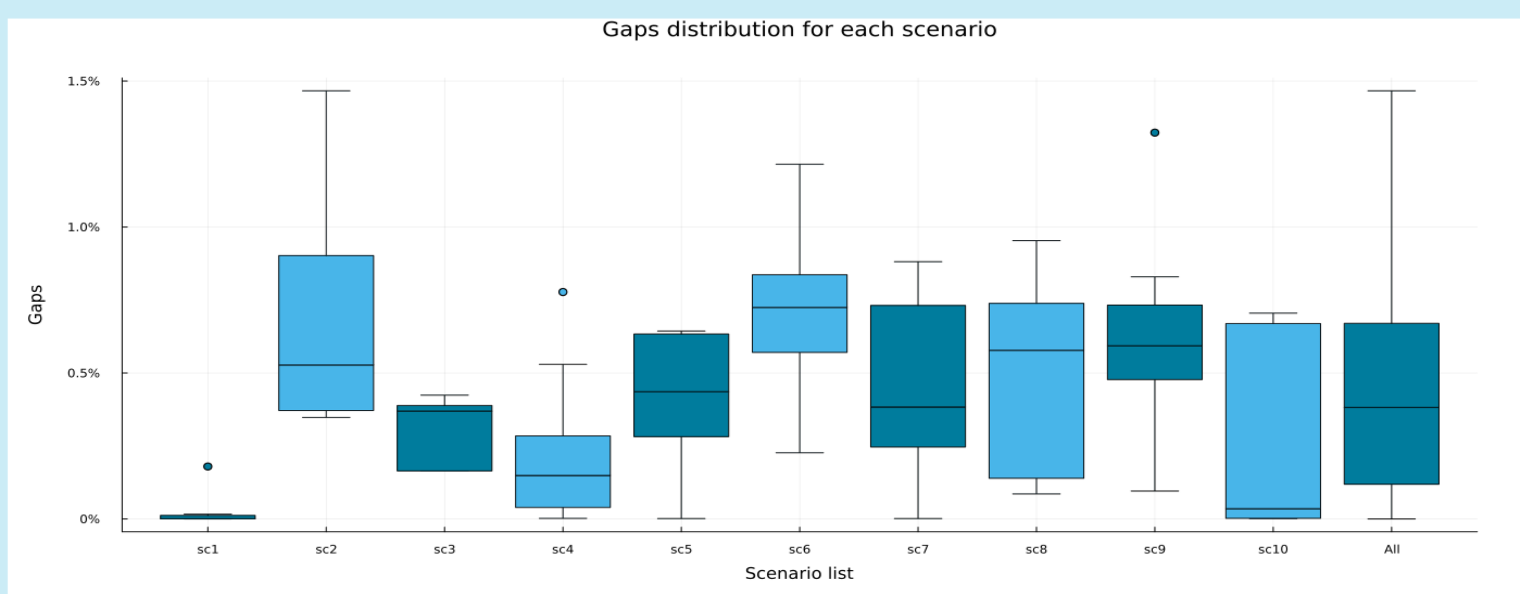


Fig 2: Representation of a one Manhattan scenario solution. Green markers denote open stations, while red markers signify closed stations.



References

[1] :Çalık, Hatice, and Bernard Fortz. "Location of stations in a one-way electric car sharing system." 2017 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2017.